

Map Library Usage

There are several supported map libraries available to take advantage of NASA's Global Imagery Browse Services (GIBS) via OGC Web Map Tile Service (WMTS), Tiled Web Map Service (TWMS), and KML files. Listed below is a set of map libraries, code snippets, working examples, and screen captures designed to help in building your own user interface to explore NASA's Earth imagery. Alternatively, these guidelines can also help to integrate that imagery into your existing client or to build scripts to retrieve imagery from GIBS. Geographic Information System (GIS) users: [go here for instructions](#).

Code Available in GitHub

In addition to the snippets below, fully-working code examples are available for OpenLayers and Leaflet in the [GIBS GitHub project area](#).

Web-based Libraries

OpenLayers 3

[OpenLayers 3](#) is an open source, web browser-based mapping library written in JavaScript. OpenLayers maps can also be viewed on mobile devices since no plugins are required to use it.

See the [Live Examples](#) (including [Geographic](#), [Web Mercator](#), [Arctic](#), and [Antarctic](#)), notes on using OpenLayer 3 with [GIBS](#), and clone the [GIBS Web Examples](#) from GitHub to tinker with the code. The [EOSDIS Worldview tool \(source\)](#) is based on OpenLayers 3, too.

Leaflet

[Leaflet](#) is an open source, web browser-based and mobile-friendly mapping library written in JavaScript.

See the [Live Examples](#) (including [Geographic](#), [Web Mercator](#), [Arctic](#), and [Antarctic](#)), notes on using Leaflet with [GIBS](#), and clone the [GIBS Web Examples](#) from GitHub to tinker with the code. For details on how to configure specific layers available from GIBS, follow the relevant instructions included in the [GIBS Layer Configuration Information](#).

Google Maps

Maps created with the [Google Maps API, version 3](#) can include layers from GIBS. See the [GIBS Web Examples](#) for sample code. For details on how to configure specific layers available from GIBS, follow the relevant instructions included in the [GIBS Layer Configuration Information](#).

Bing Maps

Maps created with the [Bing Maps AJAX Control, version 7](#) can include layers from GIBS. See the [GIBS Web Examples](#)

for sample code. For details on how to configure specific layers available from GIBS, follow the relevant instructions included in the [GIBS Layer Configuration Information](#).

Google Earth (browser plugin)

Google Earth (Plugin) is a virtual globe that can be embedded on a webpage and manipulated through Javascript. It utilizes the [KML interface of TWMS](#) to retrieve data.

Requirements

- Google Earth Plugin

Limitations

Google Earth does not have a convenient method of layering maps on top of each other. One can achieve the effect by changing the elevation that the map layers are displayed at, although that sometimes can introduce unwanted artifacts on the display.

Usage

The Javascript uses a callback function to load data from a URL. The list of available products can be found [here](#).

```
var url =
"https://gibs.earthdata.nasa.gov/twms/epsg4326/best/kmlgen.cgi?layers=MODIS_Terra_CorrectedReflectance_TrueColor&time=2012-07-12";
google.earth.fetchKml(ge, url, function(object){
  if (!object) {
    // wrap alerts in API callbacks and event handlers
    // in a setTimeout to prevent deadlock in some browsers
    setTimeout(function() {
      displayDialog('Bad or null KML.');
    }, 0);
    return;
  }

  ge.getFeatures().appendChild(object);
  object.setVisibility(true);
  //object.setOpacity(alpha); // use to change transparency
  //of layer
});
}
```

Screenshots

Unknown Attachment

GIBS data displayed in the Google Earth Plugin (As used in PO.DAAC's State of the Ocean when "MODIS True Color" layers are selected)

Additional Documentation

- [Google Earth Plugin API](#)

Desktop Application Libraries

NASA World Wind

World Wind is a multi-platform virtual globe developed by NASA that runs in Java. It can load GIBS data through either the [KML interface](#) or the [TWMS interface](#).

Requirements

- NASA World Wind 1.3+ or the [online JNLP demos](#)
- Java 1.6+

Usage (KML)

To run the World Wind client with [GIBS KML capabilities](#), load the "KML Viewer" demo application from either the standalone client or the [online demo](#). To load a specific layer, select File | Open URL... and enter a URL for the layer in this format: `https://<url to kmlgen.cgi>?layers=<layername>&time=<time>` for example, https://gibs.earthdata.nasa.gov/twms/epsg4326/best/kmlgen.cgi?layers=MODIS_Terra_CorrectedReflectance_TrueColor&time=2012-06-21

To enable time adjustment on a particular layer, use the following URL parameter: `time=R10/<date>/P1D` for example, https://gibs.earthdata.nasa.gov/twms/epsg4326/best/kmlgen.cgi?layers=MODIS_Terra_CorrectedReflectance_TrueColor&time=R10/2012-05-26/P1D

Additional documentation is available [here](#).

Limitations: The name of the layer has to be known prior to entering the URL. The list of available layers can be found [here](#).

Usage (TWMS)

To run the World Wind client with TWMS capabilities, load the WMS Layer Manager demo application from either the standalone client or the [online demo](#). In the layers menu, add the [TWMS endpoint](#). This will load the list of layers

available from TWMS and selecting a layer will allow it to be shown on the globe.

Limitations: World Wind will continue to zoom and try to load images even if they're beyond the depth of the highest resolution tile. When this happens, the images will go blank. In addition, this interface does not allow the selection of variables such as time.

Screenshots

There are no images attached to this page.

Google Earth (Desktop)

See the Google Earth section on the [GIBS GIS Usage page](#).

Mobile App Libraries

ESRI iOS Client Library

The iOS client library is a native library written in Objective-C and can be incorporated into any iOS app for the iPod Touch, iPhone, and iPad. It is built upon ArcGIS's iOS client libraries, which provides a MapView and asynchronous tile loading, coupled with a custom TWMS data loader.

The library comes with a sample client to display TWMS maps and includes basic manipulations such as presenting data from arbitrary TWMS server, selecting a time-value, and map layer reordering.

Requirements

- iOS 5.0+
- ArcGIS iOS Client Library 2.2.1+ ([Download here](#))

Usage

To use the library, create an AGSMapView object, either programmatically or linked from Interface Builder, and use the following code snippet as an example.

```
TWMSTiledWebMapService *earthTwms =
[ [TWMSTiledWebMapService alloc] initWithUrl:[NSURL
URLWithString:@"https://gibs.earthdata.nasa.gov/twms/epsg43
26/best/twms.cgi?request=GetTileService" ]];
TWMSTiledMap *aquaMap = [[earthTwms tiledMaps]
valueForKey:@"MODIS AQUA tileset"];
TWMSTiledMapLayer *aquaMapLayer = [[TWMSTiledMapLayer
alloc] initWithTiledMap:aquaMap];
[agsMapView addMapLayer:aquaMapLayer withName:[aquaMap
name]];
```

Screenshots

There are no images attached to this page.

Additional Documentation

The ArcGIS iOS documentation can be obtained [here](#).

The standard iOS documentation can be viewed [here](#).

The iOS library defines 3 classes that are used to load TWMS data:

TWMSTiledWebMapService: Instantiated for each TWMS endpoint.

TWMSTiledMap: Represents an individual map layer in the TWMS.

TWMSTiledMapLayer: A subclass of ArcGIS's map layer to show TWMS data.

TWMSTiledWebMapService

Properties

name: The name of the TWMS as defined in the GetTiledService request.

title: The title of the TWMS as defined in the GetTiledService request.

abstract: The abstract of the TWMS as defined in the GetTiledService request.

tiledMaps: A dictionary where the keys are the name of each map layer and values are the TWMSTiledMap instances.

Methods

*(id)initWithUrl:(NSURL *)url*: Loads the TWMS data given by the url to instantiate this class.

*(NSArray *)listAllTiledMapsByName*: Returns a sorted list of map layer names.

*(void)setTiledMapKeyValue:(NSString *)value forTiledMapKey:(NSString *)key*: Assigns variable keys for all TiledMaps in the TWMS server.

*(NSDictionary *)tiledMapKeyValues*: Returns a dictionary of all the keys used in all TiledMaps and the value its currently set to.

TWMSTiledMap Properties

name: The name of the layer as defined in the GetTiledService request.

title: The title of the layer as defined in the GetTiledService request.

uuid: A randomly generated unique ID for the layer.

Code Snippets

Create a new TWMSTiledWebMapService using a URL

```
TWMSTiledWebMapService *earthTwms =  
[[TWMSTiledWebMapService alloc] initWithUrl:[NSURL  
URLWithString:@"https://gibs.earthdata.nasa.gov/twms/epsg43  
26/best/twms.cgi?request=GetTileService"]];
```

List all map layers in a TWMS server

```
NSArray *tiledMapNames = [earthTwms  
listAllTiledMapsByName];  
for(NSString *tiledMapName in tiledMapNames) {  
    NSLog(@"%@", tiledMapName);  
}
```

Update variables for all TWMS endpoints

```
[earthTwms setTiledMapKeyValue:@"2012-07-12"  
forTiledMapKey:@"${time}"];
```

Add, reorder, and remove map layers (Provided by ArcGIS's MapView class)

```
AGSMapView *mapView = <the map view>;  
TWMSTiledMap *tiledMap = <obtained from  
TWMSTiledWebMapService>;  
TWMSTiledMapLayer *tiledMapLayer = [[TWMSTiledMapLayer  
alloc] initWithTiledMap:tiledMap];  
  
[mapView addLayer:tiledMapLayer withName:[tiledMap uuid]];  
[mapView insertMapLayer:tiledMapLayer withName:[tiledMap  
uuid] atIndex:<index>];  
[mapView removeLayerWithName:[tiledMap uuid]];
```

Set map layer transparency (Provided by iOS UIView's transparency settings)

```
UIView *view = [[[mapView mapLayerViews]
objectForKey:[tiledMap uuid]]];
[view setAlpha:<transparency value>];
```

Script-level Access

GDAL

See the GIBS API documentation for [script-level access to GIBS via GDAL](#).